

## Organisations and Data Management

### 1 Data Collection: Why organisations & individuals acquire data & supply data via websites

#### Techniques used by organisations to acquire data & reasons for choice

### 3 Techniques used by organisations to protect the rights of individuals and organisations supplying data

## Data Collection: Why organisations & individuals acquire data & supply data

- Prior to online collection data collected by forms prone to mistakes
- **Reasons why organisations acquire data via websites: (using direct data collection).**
  - 24 hour customer access, no double handling forms
  - Improved efficiencies through direct data entry by customers, (no waiting for forms, accurate, etc.)
  - Improvements in effectiveness, (no poor handwriting, clear & readable, inbuilt mechanisms ensure data accuracy)
  - Access & purchase goods from overseas, (different time zones no problem)
  - Potential problem: customer inputting incorrect data deliberately

## Why individuals and organisations supply data by websites:

1. Purchasing of goods and services online
  1. Individuals house bound, elderly, time poor people, online groceries, etc.
2. Feedback
  1. Feedback relating to products, services, research, etc
3. Online voting
  1. Method of expressing an opinion
4. Social networking
  1. These sites facilitate the way in which individuals communicate with friends & family, type of data entered include email addresses, name, age, location photos, etc.

## Techniques used by organisations to acquire data & reasons for choice

### ■ Data acquisition software

#### 1 PHP

- Hypertext pre-processor, general purpose scripting language used on web servers
- Versatile, can operate on many web servers, OS, and used with any RDMS and is free
- Used to create dynamic webpage content, eg. Google Earth, Maps, BOM, search engines.
- Dynamic webpages are more flexible.
- Allows the user to enter data to access text or images

## Techniques used by organisations to acquire data

- Data acquisition software
- 2 ASP, active server pages; ASP.Net is a web application tool that assists in building dynamic webpages
  - PHP & ASP used primarily on the server side
- 3 JavaScript mainly used on the client side
- 4 Back-end tools
  - Tools to collect statistics on visitors to websites, eg. which browsers used their site, information searched for, location of visitors
  - Cpanel and Google Analytics provide organisations with this information

## Techniques used by organisations to acquire data

### ■ Data acquisition software

#### ■ 5 Cookies

- Small file that a web server stores on user's computer
- Contain data about the user, eg. email address and web-viewing preferences
- Cookie sent to computer when website is browsed and stored on the computer's hard disk
- Cookies can be used to track people which could lead to privacy issues

**Techniques used by organisations to protect the rights of individuals and organisations supplying data**

**1. Security protocols**

- TLS (Transport layer security and SSL) protocols used to provide security for communications on the internet
- HTTPS, (hypertext transfer protocol secure); combination of HTTP and SSL/TLS protocol
- HTTPS provides encryption and security in terms of identification of the server, eg. financial transactions
- HTTPS relies on certificates issued by authorities and installed on the browser software
- URLs using HTTPS begin with https:// and use port 443
- URLs using HTTP is not secure and begin with http:// and use port 80.

**Techniques used by organisations to protect the rights of individuals and organisations supplying data**

**2 Privacy Policies**

- Organisations collecting data on individuals have privacy policy on how it uses the data collected and to whom it will disclose it
- By law, privacy policies must be located on a company's website and easy to find
- Policy must include details about what data is gathered and how it is used

**Techniques used by organisations to protect the rights of individuals and organisations supplying data**

**3 Shipping and returns policy**

- Organisations selling goods & services online have a "shipping and returns policy" for customers not satisfied with the order
- Policy provides guidance on how to return the item through the postal system and necessary data required to identify the order.

## U3O2: Structure & Role of Relational Databases

### Flat File Vs Relational Database

- Refer to the *software* as "database management system (DBMS)"
  - E.g. Access, Filemaker, OpenOffice Base.
- Refer to the product created with the software as a "database".
  - E.g. student records, music collection.

1

### Structure & Role of Relational Databases

- A primary key:
  - A field attached to each record in a database. The value of this key should be unique for each record in the database
  - Egs. number plates on cars, TFN's, student numbers.
- Form, a form allows an input screen to be formatted and linked to an underlying table or query.
- Query, to filter a set of data
- Report, formats the query data and enables summary statistics to be added
- Macros, carry out a set of predetermined tasks, eg. print

2

### Structure & Role of Relational Databases

#### Data Formats:

- text
- fixed decimal places
- Date
- Date/time
- Dollar, currency
- True/false, Boolean logic

• Data Types: Text, (string) String data types include a series of symbols or values, such as a character string (a sequence of characters) or a binary string (a sequence of binary values)

3

### Structure & Role of Relational Databases

#### Data Types:

| Data type | Characteristics or uses   | Examples   |
|-----------|---|--|
| Text      | Alphanumeric; up to 255 characters; it is searchable                          | Name, address, postcode, telephone number  |
| Numeric   | Numbers only (see below for different formats)                                | Any number that will be used in a calculation  |
| Currency  | Numbers, but in dollar amounts, formatted with \$ symbol and .00 as a default | Any number used to represent a financial value; usually only applied to a total, rather than value in a list |
| Date/Time | A variation of numbers, but formatted to represent a date and/or time         | Any date; can be used in calculations  |
| Boolean   | Represents one of two states - True/False                                     | Also represented as Yes/No and On/Off  |
| Object    | Images, audio, video or similar   | Media or other documents   |
| Memo      | Like text, but unlimited; it is not   | Any long selections of text  |

4

### Types of Databases

- Q. So, what IS a relational database?
- A. A database with 2 or more related tables. A RDMS in addition to storing data stores data relationships between tables.

A *flat file database* has a single table and often suffers from inefficient repetition of data.

*Normalisation* is the process of removing this repetition by crafting relationships between tables.

5

### Flat file

| Name   | Department | Boss          | Phone    |
|--------|------------|---------------|----------|
| Smith  | Sales      | Britney Lurgi | 0123 456 |
| Jones  | Sales      | Britney Lurgi | 0123 456 |
| Lennon | Sales      | Britney Lurgi | 0123 456 |
| Sade   | Transport  | Tom Brick     | 0876 543 |
| Masoch | Transport  | Tom Brick     | 0876 543 |

### Relational

| STAFF TABLE |            |
|-------------|------------|
| Name        | Department |
| Smith       | Sales      |
| Jones       | Sales      |
| Lennon      | Sales      |
| Sade        | Transport  |
| Masoch      | Transport  |

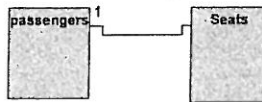
  

| DEPARTMENTS TABLE |               |          |
|-------------------|---------------|----------|
| Department        | Boss          | Phone    |
| Sales             | Britney Lurgi | 0123 456 |
| Transport         | Tom Brick     | 0876 543 |

- + No unnecessary data repetition
- + Department info easily modified in one operation.

## Types of Relationships, connection between data

- One-to-one relationship:
- Eg. Airline's passenger details:



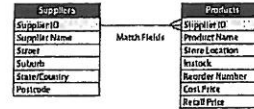
Each passenger has only one seat and each seat can be assigned to only one passenger.

A record in one table is connected to only one record in a second table.

7

## Types of Relationships, connection between data

- One-to-many relationship:
- Eg. customer details & job quotation:



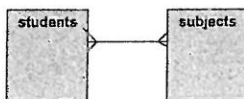
The one supplier can provide MANY products, but each product comes from ONE supplier.

Also, eg. several workers share single telephone extensions; each extension record is related to several employee's records.

8

## Types of Relationships, connection between data

- many-to-many relationship:
- Eg. Student details and subjects tables:

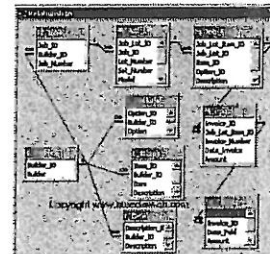


Each student studies many subjects and each subject has many students

9

## Advanced

- A multi-table real-life database such as a commercial system with tables for ...
  - Client data
  - Supplier data
  - Inventory
  - Sales



10

## Determining a RDBMS structure

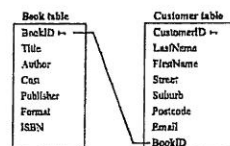
- Need to determine which field in each database will be the **Primary Key**, a value which is unique for each record, eg. customer ID no)
- Foreign keys**, fields are used to ensure that if you are entering data in one table, it already has a corresponding value in another
- Used to create relationships between tables
- Used to enforce referential integrity by ensuring relationships between tables remain consistent
- No uniqueness constraint for a foreign key

11

## Determining a RDBMS structure

- Foreign keys:**

The design for a small bookshop's relational database links the BookID in both the Book and Customer tables.



The Book ID field in the customer table is a foreign key.

12

### Determining a RDBMS structure Table Normalisation

- Table normalisation
  - Process of efficiently organising data in a database
  - Use of a set of rules to check for anomalies or deviations in data structure to ensure fields are in the correct tables
  - Process:
    - Eliminates redundant data, (eg. storing the same data in more than one table); saves lots of storage space and speeds up data access
    - Changes need only be made in one place rather than in many places.

13

### Determining a RDBMS structure Table Normalisation

- Table normalisation
  - Changes need only be made in one place rather than in many places.
  - Ensures data dependencies make sense, (only storing related data in a table).
  - More powerful data access is possible

14

### Determining a RDBMS structure Table Normalisation

- Table normalisation
  - There are six "normal forms", each rule applied successively from
  - First normal form, (1NF).
  - Eliminated duplicative columns from same table
  - Only one value, not a list of values in cell
  - Create separate tables for each group of related data & identify each row with a unique primary key

| ProductID | ProductName | SellingPrice |
|-----------|-------------|--------------|
| 10490     | Sausage     | 1.00         |
| 87653     | Hamburger   | 3.50         |
| 44012     | Lemonade    | 1.25         |
| 80432     | Chocolate   | 1.95         |

15

Your turn... repair this!

|     |            |           |
|-----|------------|-----------|
| 111 | Fred Smith | 4568 3456 |
| 222 | Mary Jones | 4567 8900 |
| 333 | Tim Blogs  | 3254 5676 |

Repaired!

|     |      |       |
|-----|------|-------|
| 111 | Fred | Smith |
| 222 | Mary | Jones |
| 333 | Tim  | Blogs |

Now, customers can be sorted and searched by first name and/or surname separately.

Also, the names can be used individually, like "Dear Fred" instead of "Dear Fred Smith"

Repair This!

|      |       |       |
|------|-------|-------|
| A345 | Red   | 4kg   |
| A568 | Blue  | 300g  |
| B695 | White | 1.5kg |

### Repaired!

| A345 | Red   | 4000 |
|------|-------|------|
| A568 | Blue  | 300  |
| B695 | White | 1500 |

### Repair This!

| Monster           | 1 | 3:23 |
|-------------------|---|------|
| Monster           | 2 | 4:12 |
| Collapse into Now | 1 | 4:01 |

### Repaired

| Monster           | 1 | 203 |
|-------------------|---|-----|
| Monster           | 2 | 252 |
| Collapse into Now | 1 | 241 |

- Time notation like "3:23" represents two pieces of data – minutes and seconds – that mean nothing to a database
- Cannot be understood a database without serious text parsing
- Single "seconds" value can be sorted, searched, compared

### Repair This!

| 111 | 66 Lake Rd, Mentone, 3198          |
|-----|------------------------------------|
| 222 | 2/45 Richmond Lane, Richmond, 3121 |
| 333 | 135 Spring St, Melbourne, 3000     |

- And address like "3 Fred St, Sale, 3586" has 3 pieces of data: street address, town, postcode.

### Repaired!

| 111 | 66 Lake Rd         | Mentone   | 3198 |
|-----|--------------------|-----------|------|
| 222 | 2/45 Richmond Lane | Richmond  | 3121 |
| 333 | 135 Spring St      | Melbourne | 3000 |

Now each field can be searched & sorted and used individually (e.g. addressing envelopes)

### Repair this... it's tricky!

| Customer ID | Name       | Phone                                  |
|-------------|------------|--|
| 111         | Fred Smith | 4568 3456                              |
| 222         | Mary Jones | 4567 8900<br>(BH)<br>3456 2314<br>(AH) |
| 333         | Tim Blogs  | 3254 5676<br>0402 697 495              |

## First attempt...

| ID  | Name       | Phone                  |
|-----|------------|------------------------|
| 111 | Fred Smith | 4566 3456              |
| 222 | Mary Jones | 4567 8900 3456 2314    |
| 333 | Tim Blogs  | 3254 5676 0402 697 495 |

### Problems:

- Trouble querying the table: "Which customer has phone # 3456 2314?" Have to search more than 1 field... messy, Ugly.
- Can't enforce validation rules to prevent duplicate phone #'s
- Can't enter three or more phone numbers
- Waste of space for all people with only 1 number

## Second attempt...

| Customer ID | Name       |
|-------------|------------|
| 111         | Fred Smith |
| 222         | Mary Jones |
| 333         | Tim Blogs  |

| Customer ID | Phone        |
|-------------|--------------|
| 111         | 4566 3456    |
| 222         | 4567 8900    |
| 222         | 3456 2314    |
| 333         | 3254 5676    |
| 333         | 0402 697 495 |

### Benefits:

- Unlimited phone numbers for everyone!
- No need to search multiple Phone fields
- No need to tear apart text from one field to extract a particular number
- All we need is a 1:many relationship between customer name table and customer phone table using the ID as the key field.

## Determining a RDBMS structure Table Normalisation

- Table normalisation
  - Second Normal Form, (2NF)
  - Meet all requirements of the first normal form
  - Remove subsets of data that apply to multiple rows of a table and place them in separate tables
  - Following table doesn't comply with 2NF as if student changes form the occurrence of form would have to be updated for each subject.

| Name | Subject     | Form |
|------|-------------|------|
| 111  | IT          | 111  |
| 111  | History     | 111  |
| 111  | English     | 111  |
| 222  | IT          | 222  |
| 222  | English     | 222  |
| 222  | Mathematics | 222  |
| 333  | IT          | 333  |

27

## Determining a RDBMS structure Table Normalisation

- Table normalisation
  - Second Normal Form, (2NF)
  - Using 2 tables to create a 2NF structure

| Name | Form | Name | Subject     |
|------|------|------|-------------|
| 111  | 111  | 111  | IT          |
| 111  | 111  | 111  | History     |
| 111  | 111  | 111  | English     |
| 222  | 222  | 222  | IT          |
| 222  | 222  | 222  | English     |
| 222  | 222  | 222  | Mathematics |
| 333  | 333  | 333  | IT          |

28

## 2NF – Second Normal Form

- Removes subsets of data that apply to multiple rows of a table and places them in separate tables.
- Creates relationships between these new tables and their predecessors using foreign keys.

## 2NF example

| ID  | Name       | Phone     |
|-----|------------|-----------|
| 111 | Fred Smith | 1293 5934 |
| 222 | Mary Jones | 4839 2712 |
| 222 | Mary Jones | 2849 5967 |
| 333 | Ike Turner | 2393 4955 |

If Mary Jones got married and changed her name, changes would need to be made in more than one record. If one change were missed, the integrity of the data would be damaged.

Making multiple changes like this is also time-consuming and repetitious, thereby eating up storage space.

Solution: Store names only once in a separate table, as in the phonenum example before. Name changes now only need to be made once.

## Solution

| Customer ID | Name       |
|-------------|------------|
| 111         | Fred Smith |
| 222         | Mary Jones |
| 333         | Tim Blogs  |

| Customer ID | Phone        |
|-------------|--------------|
| 111         | 4566 3456    |
| 222         | 4567 8900    |
| 222         | 3456 2314    |
| 333         | 3254 5676    |
| 333         | 0402 697 495 |

### Without NF2: flat file

| Name     | Department | Job       | Salary |
|----------|------------|-----------|--------|
| Albert   | Sales      | Manager   | 320    |
| Betty    | IT         | Analyst   | 210    |
| Clarence | Transport  | Driver    | 180    |
| Doreen   | Transport  | Driver    | 180    |
| Evelyn   | Sales      | Assistant | 240    |
| Fred     | IT         | Analyst   | 210    |
| George   | Transport  | Driver    | 180    |
| Helen    | Management | Analyst   | 210    |
| Jackie   | Sales      | Assistant | 240    |
| Jane     | IT         | Analyst   | 210    |
| Ken      | Transport  | Driver    | 180    |
| Larry    | Sales      | Assistant | 240    |
| Mary     | IT         | Analyst   | 210    |
| Nancy    | Management | Analyst   | 210    |
| Paul     | Management | Analyst   | 210    |

### With NF2: relational

| Name     | Department | Job       | Salary |
|----------|------------|-----------|--------|
| Albert   | Sales      | Manager   | 320    |
| Betty    | IT         | Analyst   | 210    |
| Clarence | Transport  | Driver    | 180    |
| Doreen   | Transport  | Driver    | 180    |
| Evelyn   | Sales      | Assistant | 240    |
| Fred     | IT         | Analyst   | 210    |
| George   | Transport  | Driver    | 180    |
| Helen    | Management | Analyst   | 210    |
| Jackie   | Sales      | Assistant | 240    |
| Jane     | IT         | Analyst   | 210    |
| Ken      | Transport  | Driver    | 180    |
| Larry    | Sales      | Assistant | 240    |
| Mary     | IT         | Analyst   | 210    |
| Nancy    | Management | Analyst   | 210    |
| Paul     | Management | Analyst   | 210    |

- Department data is only stored **once**. So:
- Less storage space required
  - Department changes now only made **once**, not once for *each worker in that dept*

## 2NF

| Manufacturer | Model      | Model Full Name      | Manufacturer Country |
|--------------|------------|----------------------|----------------------|
| Forte        | X-Prime    | Forte X-Prime        | Italy                |
| Forte        | Ultraclean | Forte Ultraclean     | Italy                |
| Dent-o-Fresh | EZbrush    | Dent-o-Fresh EZbrush | USA                  |
| Kobayashi    | ST-60      | Kobayashi ST-60      | Japan                |
| Hoch         | Toolmaster | Hoch Toolmaster      | Germany              |
| Hoch         | X-Prime    | Hoch X-Prime         | Germany              |

The table above is a problem.  
 Let's say {Model Full Name} is the primary key.  
 The {Manufacturer Country} field is based on the {Manufacturer} field, and will need to be constantly updated if manufacturers change their location.  
 To be properly 2NF, you'd need to do this...

## 2NF

| Manufacturer | Model      | Model Full Name      | Manufacturer Country |
|--------------|------------|----------------------|----------------------|
| Forte        | X-Prime    | Forte X-Prime        | Italy                |
| Forte        | Ultraclean | Forte Ultraclean     | Italy                |
| Dent-o-Fresh | EZbrush    | Dent-o-Fresh EZbrush | USA                  |
| Kobayashi    | ST-60      | Kobayashi ST-60      | Japan                |
| Hoch         | Toolmaster | Hoch Toolmaster      | Germany              |
| Hoch         | X-Prime    | Hoch X-Prime         | Germany              |

## 2NF

Break the data into two tables

| Manufacturer | Manufacturer Country |
|--------------|----------------------|
| Forte        | Italy                |
| Forte        | Italy                |
| Dent-o-Fresh | USA                  |
| Kobayashi    | Japan                |
| Hoch         | Germany              |
| Hoch         | Germany              |

| Model      | Model Full Name      |
|------------|----------------------|
| X-Prime    | Forte X-Prime        |
| Ultraclean | Forte Ultraclean     |
| EZbrush    | Dent-o-Fresh EZbrush |
| ST-60      | Kobayashi ST-60      |
| Toolmaster | Hoch Toolmaster      |
| X-Prime    | Hoch X-Prime         |

## 2NF

Make the same key fields in each table

| Manufacturer | Manufacturer Country |
|--------------|----------------------|
| Forte        | Italy                |
| Forte        | Italy                |
| Dent-o-Fresh | USA                  |
| Kobayashi    | Japan                |
| Hoch         | Germany              |
| Hoch         | Germany              |

| Model      | Model Full Name      |
|------------|----------------------|
| X-Prime    | Forte X-Prime        |
| Ultraclean | Forte Ultraclean     |
| EZbrush    | Dent-o-Fresh EZbrush |
| ST-60      | Kobayashi ST-60      |
| Toolmaster | Hoch Toolmaster      |
| X-Prime    | Hoch X-Prime         |



## 2NF

| Manufacturer | Manufacturer Country |
|--------------|----------------------|
| Forte        | Italy                |
| Forte        | Italy                |
| Dent-o-Fresh | USA                  |
| Kobayashi    | Japan                |
| Hoch         | Germany              |
| Hoch         | Germany              |

Set up the relationship between the key fields in each table

| Manufacturer | Model      | Model Full Name      |
|--------------|------------|----------------------|
| Forte        | X-Prime    | Forte X-Prime        |
| Forte        | UltraClean | Forte UltraClean     |
| Dent-o-Fresh | EZbrush    | Dent-o-Fresh EZbrush |
| Kobayashi    | ST-60      | Kobayashi ST-60      |
| Hoch         | Toolmaster | Hoch Toolmaster      |
| Hoch         | X-Prime    | Hoch X-Prime         |

## Determining a RDBMS structure Table Normalisation

- Table normalisation
  - Third Normal Form, (3NF)
  - Meet all the requirements of the 2NF
  - Remove columns that are not dependent upon the primary key; every field in a table must relate directly to the primary key; eg. the Event is the key. The Year and Winner's name are both directly related to the Event. If a competitor wins several events, data errors can appear if the date of birth is entered differently each time.
  - To correct this, two tables formed to comply with the 3NF.

38

## Determining a RDBMS structure Table Normalisation

- Table normalisation; Third Normal Form, (3NF)

| Event       | Year | Winner's name | Winner's date of birth |
|-------------|------|---------------|------------------------|
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |
| 100m Sprint | 2003 | Armed         | 10 Mar 99              |

WARNING: This table is not 3NF because the date of birth is not directly related to the primary key.

| Event       | Year | Winner's name | Computer | Date of birth |
|-------------|------|---------------|----------|---------------|
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |
| 100m Sprint | 2003 | Armed         | Compu    | 10 Mar 99     |

39

## Normalisation

Can someone please help me with this Sample Exam question. I am trying to understand table normalization. I have a general understanding of what it means, but I'm not really sure about the actual differences in this particular question between the second and third normal forms. As a class we have been going through the textbook definition, but I'm struggling to apply it to this particular sample question.

Also if anyone has some good resources on a simple (not so technical way) of how to explain this easier it would be greatly appreciated.

Thanks

### Question 1

Below is a table from a yacht club's relational database.

| Name          | Boat Name | Address_Suburb            | State | memType | Fees \$ | Gender | Boat Storage Fees \$ | Total Fees Due \$ |
|---------------|-----------|---------------------------|-------|---------|---------|--------|----------------------|-------------------|
| Sue Trowsdale | Suzy      | 21 Mountain Rd<br>Rosebud | VIC   | Senior  | 250.00  | F      | 125.00               | 375.00            |
| Jon Ross      | Wahoo     | 64 View Rd McCrae         | VIC   | Junior  | 140.00  | M      | –                    | 140.00            |
| Harry Wilson  | Victor    | 35 Shady Ln<br>Rosebud    | VIC   | Life    | –       | M      | 125.00               | 125.00            |

2 marks

- a. Describe what would happen when you apply **one** of the First, Second or Third Normal Forms to data in the table. Refer to the table in your answer.

Tick the box to indicate which normal form you are describing.

First ☐ or Second ☐ or Third ☐

Nothing so interesting as discussing NF1-3, more a Panadeine Forte and a bottle of red I would have thought :-). The article I suggested was the "least confusing" of several found. There does however seem to be perhaps a need for the VCAA to spell out a bit more clearly its meanings of NF1-3 as there do seem to be several different versions around.

If you want a really clear explanation :-( then try <http://dev.mysql.com/tech-resources/articles/intro-to-normalization.html#03>

I would be interested to know the origin of Laura's sample question. And just what they did expect as an answer for two marks. Also how can you apply just one of NF1, NF2 or NF3 to a table when each one assumes the previous requirements have been satisfied?

Where do we go?

1NF: Two new fields in the database table so that the "Street Address" and "Suburb" become separate fields and "Family\_Name" and "Given\_Name" also become separate fields - atomicity, no more than one value in a field

2NF: Dividing the data into at least two tables: eg perhaps Member\_Details: Primary key, Member names, Address details, Gender (? is gender needed, if not is it allowed under present laws?), Membership type

and Membership\_Types: Primary key, Membership fee, Boat storage fee

and establishing the relationships between the tables

### **Why Normalise?**

Two reasons; eliminate redundant data and to ensure data in each table is logically related.

### **Normalisation process**

Create separate fields/tables as necessary to...

1. eliminate multiple entries in fields (eg Name into Surname and first\_name fields)
2. eliminate duplicate columns
3. remove columns that are not dependent upon the primary key

Establish relationships by...

4. ensuring each table has a primary key
5. using foreign keys as necessary

Plus remember that only data is stored in tables (eg not calculations)

Now where all this fits into 1NF, 2NF and 3NF is different depending on which source you look at.

Paula C, any chance of some guidance here? Is the 1NF, 2NF and 3NF stuff going to be examined?  
If so we need some clear guidelines on what actually happens within the three Normal forms?  
(according to VCAA anyway)

thanks

Mark Scott

Luther College

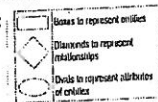
## U3O2: Database Design Tools

- Naming Conventions
- Eg.s prefix tables with tblCustomer, tblProducts; customer table, cusCustomerID, cusAddress; Queries, prefixed with qry; forms with frm; reports rpt.
- Ensure names are descriptive.
- Spaces & underscores avoided in object names

1

## U3O2: Database Design Tools

- Entity-relationship diagram, (ERD)
  - Used to establish interrelationships between different data elements
  - Use symbols
  - In an ERD entities are things about which information is sought, eg. books, films & attributes are the elements of data we collect about those entities, (eg. title and author of book)
  - Relationships show the linking to draw related data from different entities.



2

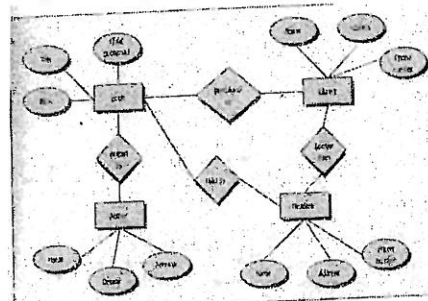
## U3O2: Database Design Tools

- Entity-relationship diagram, (ERD)
- To create an ERD:
  - Identify the entities
  - Define the relationships
  - Add the attributes to each entity

3

## U3O2: Database Design Tools

- Entity-relationship diagram, (ERD)



4

## U3O2: Database Design Tools

- Data Structure Table: Note: database tables cannot hold formulas.

| Field        | Data type | Field size | Input mask     | Caption    | Description                       | Validation rule       | Validation text                                |
|--------------|-----------|------------|----------------|------------|-----------------------------------|-----------------------|--|
| tblClientID  | Text      | 4          |                | Client ID  | Client's individual assigned code | Between 1000 and 9999 | Client ID number must be between 1000 and 9999 |
| tblLastName  | Text      | 30         |                | Last name  | Client's last name                |                       |  |
| tblFirstName | Text      | 30         |                | First name | Client's first name               |                       |  |
| tblAddress   | Text      | 60         |                | Address    | Client's street address           |                       |  |
| tblSuburb    | Text      | 30         |                | Suburb     | Client's suburb                   |                       |  |
| tblPostcode  | Text      | 10         |                | Postcode   | Client's postcode                 |                       |  |
| tblPhone     | Text      | 30         | (00) 0000 0000 | Telephone  | Client's phone number             |                       |  |

## U3O2: Database Design Tools

- Data Structure diagram
  - Indicate relationships existing between specific tables of database; indicate type of relationship in diagram.

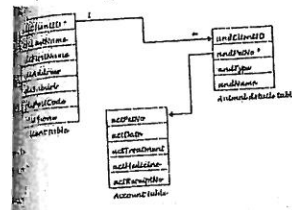


FIGURE 3-20 Data structure diagram with one-to-many relationships

6

- Specify fields to be included and the tables to which they belong
- Query criteria use symbols, plain language, \*? Wildcards, "is null", "is not null"

[illegible]

7

- Sketching of input form or output (reports)
- Location of headings, labels, fields, use of formats & conventions
- Annotated to show formatting for all elements, font type, size 7 style
- Formulas used to be shown

8

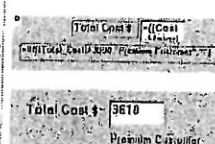
[illegible]

6

- Test data prepared in the design stage
- Test data chosen to test all aspects of solution
- Test plan is used to show all functions to be tested

| Formula/Function to test                            | Test data  | Expected/Required result   | Why?  |
|---|--|--|---|
| Min array on field<br>all cells in the same<br>date | 1/1/2000<br>8/2/2000<br>1/1/2001<br>1/1/2001<br>5/2/2001<br>6/2/2001<br>6/2/2001 | 8/2/2000<br>1/1/2001<br>1/1/2001<br>1/1/2001<br>5/2/2001<br>6/2/2001<br>6/2/2001 | The smallest "first" ID number must lie between 1/1/2000 and 5/2/2001 inclusive.<br>should be displayed |
| Sum array on field<br>all cells in the same<br>date | 1/1/2000<br>8/2/2000<br>1/1/2001<br>1/1/2001<br>5/2/2001<br>6/2/2001<br>6/2/2001 | 8/2/2000<br>1/1/2001<br>1/1/2001<br>1/1/2001<br>5/2/2001<br>6/2/2001<br>6/2/2001 | Both values are outside the range of 1/1/2000 to 5/2/2001 inclusive                                     |
| Sum array on field<br>all cells in the same<br>date | 1/1/2000<br>8/2/2000<br>1/1/2001<br>1/1/2001<br>5/2/2001<br>6/2/2001<br>6/2/2001 | 8/2/2000<br>1/1/2001<br>1/1/2001<br>1/1/2001<br>5/2/2001<br>6/2/2001<br>6/2/2001 | Worked out on a calculator  |

- Use of manual and electronic measures
- Electronic measures include validation rules
- Input masks , eg, a 9 allows only any single digit number to be entered



**FIGURE 3-35**  
An example of an IIF() statement (a) in Design view and (b) the resulting message. Note that this IIF() statement is being used to flag data in a field, rather than to validate it.

1

[illegible]

FIGURE 3-75

[illegible]

12

## U3O2: Developing a RDMS

- Steps in developing database:
  - Creating the tables and relationships
  - Setting up the queries
  - Input forms
  - Creating the macro

1

## U3O2: Developing a RDMS Testing the solution

- Informal testing, occurs during the development phase of PSM.
  - Constantly testing to see if solution behaves according to expectations
- Formal testing, testing overall solution after it has been completed
  - Formalised with comprehensive documentation of each test
- Bench test, part of formal testing
  - Sample data used to determine how solution behaves

2

## U3O2: Developing a RDMS Testing the solution

- Bench test, contd.
  - Use of a test plan created during design phase is used
  - Results of testing written on printouts from solution
- User acceptance testing
  - Asking users to follow a series of steps to complete a task in the solution or interpreting information contained in the output and then providing feedback to the developers

3

## U3O2: Developing a RDMS Testing the solution

- Features to be tested in a database:
  - Table, validation, input masks
  - Query, filter, sorting, formulas
  - Form
  - Report
  - Macro
  - User acceptance

4